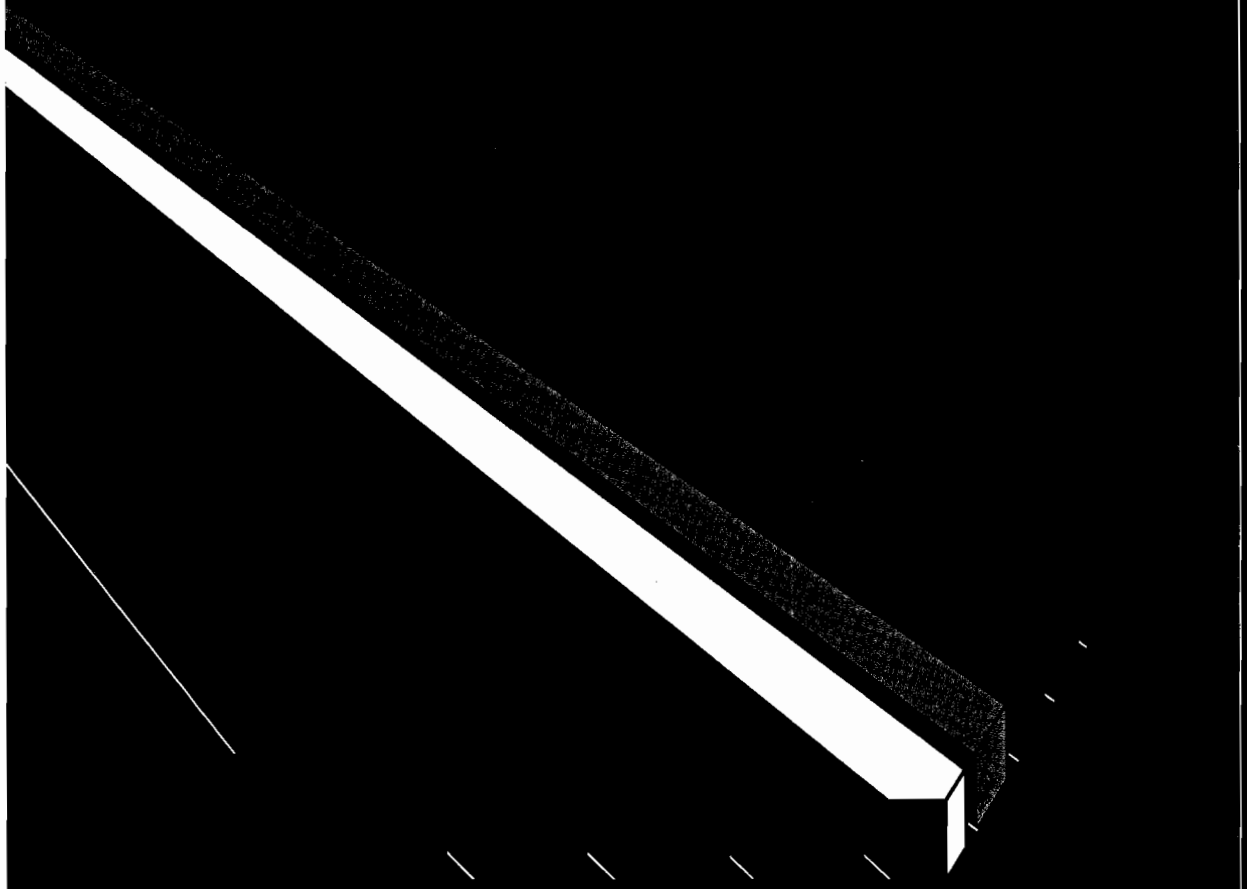
intel

Guide to Configuring Multibus®-Based Systems



intel

USER'S GUIDE TO CONFIGURING MULTIBUS®-BASED SYSTEMS

Intel Corporation, 5200 NE Elam Young Parkway, Hillsboro, OR 97123

Order Number: 144788-001

REV.	REVISION HISTORY	PRINT DATE
-001	Original Issue	5/82

Additional copies of this manual or other Intel literature may be obtained from:

Literature Department
Intel Corporation
3065 Bowers Avenue
Santa Clara, CA 95051

The Information in this document is subject to change without notice.

Intel Corporation makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Intel Corporation assumes no responsibility for any errors that may appear in this document. Intel Corporation makes no commitment to update nor to keep current the information contained in this document.

Intel Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in an Intel product. No other circuit patent licenses are implied.

The following are trademarks of Intel Corporation and its affiliates and may be used only to identify Intel products:

BXP	Intel	iSBX	Multichannel
CREDIT	Int$_e$l	Library Manager	Multimodule
i	Intelevision	MCS	Plug-A-Bubble
ICE	Intellec	Megachassis	PROMPT
iCS	iOSP	Micromainframe	RMX/80
im	iRMX	Mircromap	System 2000
iMMX	iSBC	Multibus	UPI
Insite	iPDS		

PREFACE

The Multibus interface, developed by Intel in 1977, was the first multiprocessing bus architecture to have its physical and electrical specifications released to the public. Because the Multibus architecture provided well-documented specifications along with versatility and ease of use, the electronics industry quickly accepted the Multibus architecture as a default industry standard. In fact, the Multibus architecture has become so well-accepted that over 100 manufacturers now produce Multibus-compatible boards and backplanes.

Between 1977 and the present, Intel engineers have helped many customers solve problems that arose during the process of Multibus configuration. The purpose of this user's guide is to provide you with information that will help you avoid these problems.

This guide consists of two chapters:

- Chapter 1 discusses the most-common configuration mistakes and tells you how to avoid them. This chapter is very useful for anybody who configures Multibus-based systems.

- Chapter 2 explains how to avoid several noise-inducing phenomena that appear only under special circumstances. If your system's backplane is shorter than eight inches and your system does not contain any boards that use Advanced-Schottky-technology devices to drive Multibus lines, you probably will not need the information in this chapter.

Although Intel engineers have written this configuration guide to help you maximize reliability in your Multibus-based systems, you will find that the issues discussed in Chapter 2 are not unique to the Multibus architecture. On the contrary, they can affect any board or any system that you design.

Both chapters assume that you are an experienced electrical engineer, and that you are somewhat familiar with the Multibus specification (IEEE P796).

CONTENTS

FIGURES

TABLES

CHAPTER 1 AVOIDING COMMON CONFIGURATION MISTAKES

Over the last few years, many people have called Intel's customer service engineers for assistance in configuring their Multibus-based systems. During the process of helping these people, Intel's engineers realized that a handful of simple errors were causing the vast majority of the configuration problems. The purpose of this chapter is to inform you of these common configuration errors so you can avoid them when you configure your Multibus-based systems.

This chapter is organized into seven major sections, each of which discusses one of the most common errors in configuring a Multibus-based system. The sections are:

- Deciding what boards occupy what slots
- Resolving bus priority
- Generating BCLK/ and CCLK/ signals
- Avoiding accidentally overlapped memory
- Using time-out jumpers
- Delayed processing of interrupts
- Ensuring uniform compliance with the Multibus Specification

This chapter discusses the symptoms of these errors and explains how to avoid the errors before they cause your Multibus-based system any problems.

Deciding What Boards Occupy What Slots

If your Multibus-based system consists of more than one board, you must decide what board is to occupy what slot. The Multibus architecture allows a lot of freedom in this area but, for some boards, you must take special precautions. Specifically, when placing boards that generate or use either BCLK/ or CCLK/, heed the following guidelines:

- For boards using the BCLK/ signal, you should ensure that the receivers are closer to the terminators than is the generator of BCLK/. You should also place the whole collection as close to the terminators as possible.

- If your system uses the CCLK/ signal, follow the same guidelines as for the BCLK/ signal. Ensure that the receivers of CCLK/ are closer to the terminators than is the generator of CCLK/, and try to place the entire collection as close to the terminators as possible.

The reasons for this suggestion are related to signal-to-signal coupling and signal ringing. For more details, refer to Chapter 2 of this guide.

Resolving Bus Priority

The *most common mistake* in configuring a Multibus system is incorrectly specifying the technique for resolving bus priority. Symptoms of this mistake generally include the ability to perform on-board operations correctly while being unable to perform operations that require several boards. One specific example is the ability to access on-board memory while not being able to access memory on a different board. Another example is a board's inability to process bus-vectored interrupts even though the board is configured to do so.

Why Resolve Bus Priority?

When your system contains more than one master board, the master boards must compete for access to the Multibus lines. If two masters simultaneously write information to the bus, the information will be corrupted and neither master will complete its operation successfully. The purpose of resolving bus priority is to allow each Master board a period of exclusive access to the bus.

How is Priority Resolved?

Each Multibus-compatible master board contains hardware that automatically resolves bus priority. There are two techniques that this hardware can use — serial resolution, or parallel resolution. When you configure your system, you must decide which technique to use, and you must ensure that all master boards in your system use the same technique. If you fail to do this, you will encounter the symptoms described earlier.

In any Intel iSBC hardware reference manual, you can find instructions for specifying serial or parallel resolution. So this chapter does not contain these instructions. Instead, it provides you with reasons for choosing one technique or the other.

Choosing Between Serial And Parallel Resolution

Each of the two resolution techniques has advantages and disadvantages. Specifically:

- Serial Resolution
  Serial resolution involves no special hardware, but it can be used only in systems having three or fewer master boards.

- Parallel Resolution
  Parallel resolution allows a system to contain as many as 16 master boards. However, it requires that the backplane have special hardware to help resolve priority, and it is generally more complex to set up than is serial resolution.

Consequently, the guidelines are as follows. If your system contains three or fewer master boards and is not likely to require more in the future, you should probably use serial resolution. Otherwise, use parallel.

Generating BCLK/ and CCLK/ Signals

The BCLK/ signal is required in every Multibus-based system, and the CCLK/ signal is required in some. The purpose of the BCLK/ signal is to synchronize the master boards of the system. In contrast, the purpose of the CCLK/ signal is to provide a system-wide 10 Megahertz oscillator for any board to access.

If you adhere to the following guidelines, you will avoid the problems that can be caused by improper BCLK/ and CCLK/ signals:

- BCLK/

  Every Multibus-based system requires the BCLK/ signal, but only one board can drive the signal. If no boards drive it, the master boards in your system will not be able to access other boards. For example, absence of the BCLK/ signal might allow a master board to access on-board memory while not being able to access off-board memory.

  On the other hand, if more than one board generates the BCLK/ signal, several master boards may access the bus simultaneously, corrupting addresses and data.

- CCLK/

  Many Multibus-based systems do not require the CCLK/ signal. If your system does not contain any boards that require the signal, disable the signal.

  If some boards in your system need CCLK/, you must ensure that it is generated by exactly one board. If several boards generate CCLK/, the combined signal will cause boards receiving CCLK/ to malfunction. For example, multiple CCLK/ signals could lead to premature generation of XACK/ signals, which could cause data to be corrupted.

  In contrast, if CCLK/ is not generated and some board needs it, the board attempting to receive CCLK/ might fail to generate XACK/, causing another board in the system to timeout or hang.

Avoiding Accidently Overlapped Memory

Most iSBC boards containing memory allow you to designate the addresses of the on-board memory. If, while doing so, you assign one set of addresses to more than one collection of memory, you have overlapped memory. Sometimes this is done intentionally to provide "shadow" memory. But if it happens accidentally, it leads to undesirable behavior in your system.

Precise symptoms of overlapped memory are difficult to predict because they depend upon what types of memory have been overlapped. Consider the following example. Suppose that you assign the single-port, on-board memory for Master Board A to the same locations as the memory on a slave memory board. Now suppose that Master Board B attempts to access the overlapped memory. It will receive the contents of the slave memory. In contrast, if Master Board A attempts to access the overlapped memory, it will receive the contents of its on-board memory.

The most common symptom of overlapped memory is that one set of memory addresses always seems to contain incorrect information. However, other symptoms may also appear. If your system seems to access memory in an unpredictable manner, and you have already verified that the clock signals are correct and that priority resolution is correct, check the address settings for all the boards in your system. You may have overlapped memory.

Using Time-out Jumpers

Each Intel iSBC master board has a time-out (or failsafe) jumper. The purpose of this jumper is to allow you the select the behavior that you want the board to exhibit when the processor accesses a nonexistent memory address or I/O port. If the *jumper is not connected,* and the processor accesses a nonexistent memory address or I/O port, the processor will go into an infinite wait state, appearing to have ceased executing. In contrast, if the *jumper is connected,* the processor will continue running, unaware that the memory or I/O port does not exist, and unaware that any data received is invalid.

In most systems, the time-out jumper should be disconnected, ensuring that the processor does not continue running after accessing nonexistent ports or memory. However, there are situations in which the jumper should be connected.

For example, suppose that you want your system to find out what memory exists and what memory does not. You can create a program called a memory scanner that writes a fixed pattern into a word, and then reads it. If the information read differs from the information written, the memory scanner can conclude that the memory location addressed does not exist in the system. By repeating this procedure for each word in the address space, the memory scanner can find out which addresses actually have memory. This memory scanner requires that the time-out jumper be connected because, if it is disconnected, the processor will cease execution upon accessing the first missing memory location.

Consciously decide whether to connect or disconnect the time-out jumper. This jumper can be a very powerful ally if you use it knowingly. On the other hand, if you choose to ignore it, and it happens to be configured incorrectly, you might spend hundreds of man-hours trying to find the cause of your system's sudden failure or gradual corruption.You can find more-specific information about the time-out (failsafe) jumper in your iSBC board's hardware reference manual.

Delayed Processing of Interrupts

When you place your boards into your system, you must assign a priority to each master board. This allows the system to decide which board obtains next access to the bus when several boards want access simultaneously. (This is discussed in the "RESOLVING BUS PRIORITY" section of this chapter and in the hardware reference manual for your board.)

If one of your low-priority boards receives an interrupt but cannot quickly obtain access to the bus, either of two problems may arise:

1. If the board's time-out (failsafe) jumper is connected, and the board waits for bus access long enough for the time-out mechanism to send a READY signal to the processor, the processor will immediately resume running.

2. When the board finally obtains access to the bus, the interrupt signal might no longer be present.

The first case, caused by the time-out jumper being connected, generally causes failure of the system. When the processor resumes running, it reads data unrelated

to the interrupt and interprets the data as the address of the next instruction. This causes the processor to start executing instructions at some random location in memory. To eliminate this problem, simply disconnect the time-out jumper. This will allow the processor to wait as long as necessary for valid access to the bus.

In the second case, when the board finally accesses the Multibus lines and starts processing, the interrupt signal is no longer available. If the board uses an 8259 interrupt controller to process interrupts, the 8259 deems the missing interrupt as spurious and routes the interrupt to Level 7. Even if the Level 7 interrupt-servicing code checks for spurious interrupts, it cannot ascertain the level on which the interrupt actually occurred.

The best way to avoid this kind of problem is to adhere to the following guidelines:

- Recognize which boards must access the Multibus lines after an interrupt. These boards fall into two classes — those that receive bus-vectored interrupts, and those that execute instructions located in off-board memory.
- Assign these boards sufficiently high priority to ensure rapid access to the Multibus lines.
- Be careful to allow no master board to lock up the Multibus lines for excessive periods of time.

Ensuring Uniform Compliance With The Multibus Specification

Some Multibus-compatible products adhere to only a subset of the Multibus specification. You can use these products in your system, but you must ensure that the configuration of the system matches the requirements of these products. If you neglect to take time to understand the degree of compliance before using such products, you can introduce serious problems into your system. Pay special attention to the following issues.

Number of Address Lines

What is your system's address range? For example, if the processor boards in your system use 20-bit addresses, they can access a megabyte of memory. However, if one memory board in your system uses only 16-bit addresses, it assumes that the system contains no more than 64k bytes of memory.

Consider what happens when the less-capable board receives a 20-bit address from the Multibus lines. It ignores the high-order four bits of address, and always assumes that the low order 16 bits are the entire address. In other words, if this system contains more than 64K of memory, the less-capable memory board will overlap higher memory locations.

You can avoid this problem by ensuring that all boards in your system accept and generate addresses wide enough to accommodate the largest address in the system.

Number of Data Lines

Memory access capabilities must be compatible throughout the system. For example, some memory boards transfer data only a byte at a time, while other memory boards can transfer data either a byte or a word at a time. If, for example, your system uses an 8086 processor, the processor expects to be able to receive 16-bit instructions. This is not compatible with memory boards that transfer data only a byte at a time. You can avoid problems by ensuring that your memory boards are compatible with your processor boards.

Common Bus Request Signal

The common bus request (CBRQ/) signal is not supported by some older Multibus-compatible master boards. If you use new master boards in conjunction with these old master boards, the CBRQ/ signal will work properly only if you assign all of the older boards priorities higher than those of the new boards. You may, however, elect not to use the CBRQ/ signal, in which case you can configure your boards with a different set of priorities. To find out if a particular board uses CBRQ/ refer to the part of the board's hardware reference manual that describes bus arbitration.

Partial Bus-Master Capability

Some boards can serve as bus masters only when special conditions are met. The iSBC 569 and the iSBC 80/10B boards are two examples of boards that have only partial bus master capability.

If your system contains boards having only partial bus master capability, and you intend to use the boards as bus masters, your system must meet the special conditions that are spelled out in the board's hardware reference manual.

Relying on the Lock/Signal

A final consideration is the Multibus LOCK/ signal. The purpose of LOCK/ is to allow one processor board in a multiprocessor system to obtain exclusive access to off-board dual-port RAM by locking all other processors out of the dual-port RAM.

Table 1-1 shows which boards generate and receive LOCK/, and which boards have dual-port RAM. In the future, most Intel processor boards will generate LOCK/, and most Intel processor boards having dual-port RAM will receive LOCK/.

Table 1-1. Board Characteristics Relating To The LOCK/ Signal

iSBC 80/10B iSBC 80/24 iSBC 80/30* iSBC 86/12* iSBC 88/25	NEITHER GENERATE NOR RECEIVE LOCK/
iSBC 86/05	ONLY GENERATES LOCK/
iSBC 589*	ONLY RECEIVES LOCK/
iSBC 86/14* iSBC 86/30* iSBC 88/40* iSBC 88/45*	GENERATE AND RECEIVE LOCK/
Boards marked with an asterisk (*) can contain dual-port memory.	

If your system contains multiple processors, some of which require exclusive access to dual-port RAM, the easiest way to obtain such access is to use the LOCK/ signal. If you wish to use the LOCK/ signal for this purpose, you should restrict your selection of processor boards to the following two categories:

- Processors that both generate and receive the LOCK/ signal.
- Processors that only generate the LOCK/ signal but that contain no dual-port RAM.

If your system contains processor boards that are not in one of these categories, the system can still provide exclusive access to off-board dual-port RAM by using special software protocols other than the LOCK/ signal.

CHAPTER 2 NOISE-INDUCING ELECTRICAL PHENOMENA

This chapter discusses three electrical phenomena that are capable of introducing noise into your system. These phenomena are:

- Signal-to-Signal Coupling
- Ground Shifts
- Signal Ringing

Although this chapter explains each of these phenomena and tells how to reduce the amount of noise that each phenomenon can introduce, do not jump to the conclusion that all of the discussed measures are necessary. As you read this information, remember that the Multibus architecture is a very forgiving architecture in most systems. You generally need not worry about any of these phenomena.

On the other hand, if your system is failing for reasons you cannot explain, you might try some of the suggested techniques for noise reduction — especially if your system meets one or more of the following conditions:

- Contains a long backplane (longer than eight inches).
- Is lightly populated (has many empty slots to allow for future expansion).
- Has empty slots between boards.
- Contains boards that use high-speed drivers (such as Advanced Schottky devices) to drive Multibus signals.

Signal-to-Signal Coupling

When traces are adjacent, and one of the traces changes voltage, the change can adversely influence the voltage in the adjacent trace. This phenomenon is called *signal-to-signal coupling,* and the noise imparted to the adjacent trace is called *coupled noise.*

If the coupled noise is of sufficient magnitude, it can actually cause the adjacent trace to change from one logic state to another. For example, on the Multibus bus, a logic low should be no greater than 0.8 volts, and a logic high no less than 2.0 volts. (The pull-up resistor design of Multibus backplanes holds the high logic levels to 5.0 volts, providing a large DC noise margin for signals in their high state.) If two traces are adjacent and one experiences a rapid change of voltage, it can force the adjoining trace to exceed these limits.

Two physical mechanisms contribute to signal-to-signal coupling. One of these is capacitive and the other is inductive. The capacitive component of the coupling varies directly with the cross-sectional area of the adjacent trace. Because the traces have very small cross-sectional areas, the capacitive component is negligible.

In contrast, the inductive component of the signal-to-signal coupling can be very significant. The inductive component can be expressed as

$$v_n = L\, di/dt$$

where:

- L is the total mutual inductance of the parallel current paths.
- i is the current flowing in the first trace.
- t is time.

Characteristics of Signal-to-Signal Coupling

The coupled noise is directly proportional to the length of current-conducting part of the trace. This can be derived from the fact that the inductively coupled noise is directly proportional to L, which is directly proportional to the length of the traces. This means that significant signal-to-signal coupling is more likely to occur in a backplane, with long parallel traces, than it is in a board. However, if a board contains parallel traces longer than two or three inches, coupled noise on the board can be significant.

Be aware that a longer Multibus backplane *can* exhibit more signal-to-signal coupling than a shorter backplane. Note that the preceding sentence contains the word "can" rather than "does." This is because judicious placement of boards within a system can reduce the length of the trace that conducts the signal.

The coupled noise is proportional to the current, i. Signals which conduct more current (such as BCLK/ and CCLK/, with 220 ohm pull-up resistors) generally impart more coupled noise into adjacent traces.

The coupled noise is also directly proportional to the speed with which the signal is switched (di/dt). This means that a signal switched at a faster speed will impart more noise than a signal switched at a lower speed.

Finally, noise imparted by signal-to-signal coupling is additive. This means that if signals in adjacent traces rise simultaneously and fall simultaneously, the noise will be far greater than if one rises while the other falls.

Reducing Effects of Signal-to-Signal Coupling

If you adhere to the following guidelines, you can greatly reduce the amount of noise caused in your system by signal-to-signal coupling. The guidelines are presented in order of importance.

- Eliminate CCLK/ If Not Needed

  First, if your system does not require the CCLK/ signal, do not drive CCLK/ on the Multibus lines. If CCLK/ is not needed, there is no need to expose other signals to the coupled noise generated by CCLK/.

- Carefully Consider Placement of Boards

  There are three points to consider in deciding where to place your boards:

  - Reduce the length of the trace carrying the BCLK/ signal. You can do this by placing the receivers of BCLK/ close to the generator of BCLK/, and by placing the whole collection as close to the terminators as possible.

  - Reduce the length of the trace carrying the CCLK/ signal. You can do this by placing the receivers of CCLK/ close to the generator of CCLK/, and by placing the whole collection as close to the terminators as possible.

  - In general, place receivers of signals as close as possible to the boards generating the signals.

- Run BCLK/ and CCLK/ 180 Degrees out of Phase

  If your system requires CCLK/, select a board that generates CCLK/ 180 degrees out of phase with BCLK/. This will hold the maximum amount of coupled energy to approximately the same maximum as would be generated by BCLK/ alone.

- Place Boards with Faster Signal-Switching nearer Multibus Terminator

  Some boards have faster signal switching than others. If you place the boards with faster signal switching near the Multibus terminator, you can reduce the length of the current-carrying trace for these signals.

  To decide which boards have the fastest signal switching, examine your boards and note the kinds of signal drivers that each board uses. Then refer to Table 2-1 to find out which drivers provide the fastest switching.

Table 2-1. Signal Drivers Sorted From Fastest To Slowest

Fastest	Advanced Schottky
	Schottky
	Advanced Low-Power Schottky
	Low-Power Schottky
Slowest	Standard TTL

Ground Shifts

Suppose that a number of signals emanating from a single board switch simultaneously, causing a pulse of current on the ground line. This pulse causes a difference in potential between boards. This phenomenon is known as a *ground shift*. Figure 2-1 shows a model of ground-shift failure.

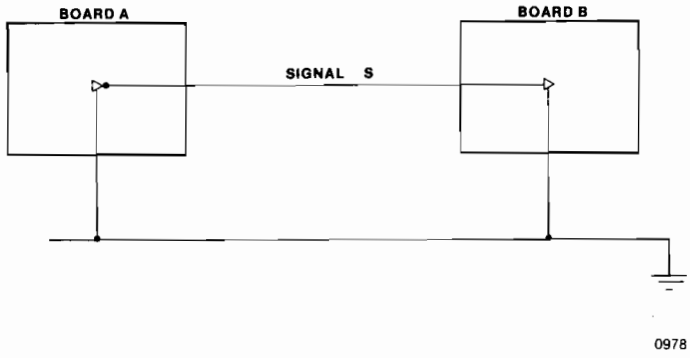


Figure 2-1. Model of Ground-Shift Failure

Adverse Effects of Ground Shift

Ground shift can have an adverse effect on the signals emanating from the boards attached to the ground lines. For example, if the ground on Board A in Figure 2-1 is at a higher potential than the ground at Board B, the potential difference is effectively added to Signal S on Board B. If the ground shift is sufficiently large, a low-level signal emanating from the Board A will be sensed as a high-level signal on Board B.

In practice, the ground shift between Boards A and B on the backplane alone is seldom sufficient to cause a problem. However, when this ground shift is added to simultaneous ground shifts within the boards, the magnitude of the ground shift can become significant.

Example of Excessive Ground Shift

For example, suppose that for a particular access across Multibus lines, the master board driving the address lines and the receiving slave (a dual-ported master with higher priority) are at opposite ends of the backplane. If current surges from the address drivers of the first master board cause a ground shift on the backplane between the two boards, this ground shift is added to the logic low outputs from the first master board. One such logic low could be the BUSY/ line. Furthermore, if a ground shift simultaneously occurs on the first master between the board-connector ground pins and bus arbiter chip (which generates BUSY/), the total ground shift

seen by the receiver of BUSY/ could equal the sum of the backplane ground shift and the on-board ground shift.

In other words, the combined ground shift could cause the second master board to think that the Multibus lines are free. This would allow the second master to use the lines at the same time as the first master, resulting in corrupted information and system failure.

Reducing the Effects of Ground Shift

Most failures relating to ground shift are the result of simultaneous ground shifts on the backplane and on a particular board. This is because the magnitude of the total ground shift is equal to the sum of the magnitudes of the individual ground shifts. Because the on-board ground shifts are of much smaller magnitude than those on the backplane, you can generally prevent system failure by reducing the magnitude of the backplane ground shift while ignoring ground shifts on individual boards.

The effect of the backplane ground shift is proportional to the distance between the board sending the signal and the board sensing the signal. Consequently, you can reduce the effect of ground shift by placing boards that communicate with each other in adjacent (or at least nearby) slots.

Examples of Reducing Effects of Ground Shift

For example, suppose that your system uses a bus master board that you suspect causes substantial ground shift. If you place this board near a higher-priority bus master, you reduce the likelihood of ground shift causing a false reading on the BUSY/ line. This can significantly improve the reliability of the interaction between the two boards.

Consider a second example. Suppose that you suspect a particular slave board of having ground-shift problems. If this board is accessed by only one master, you can improve the reliability of their interaction by placing them in adjacent slots.

Signal Ringing

When the state of a signal switches, the voltage of the signal can oscillate, as shown in Figure 2-2. This oscillation is called *signal ringing,* and its magnitude depends upon the speed of the transition from one state to another and upon whether or not the line is terminated. Specifically, the magnitude of the ringing increases as the state transition becomes faster, and the magnitude is greater when the line is not terminated.

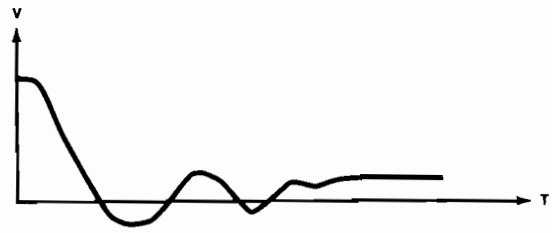


Figure 2-2. Magnitude of Signal Ringing as a Function of Time

Signal Ringing and the BCLK/ and CCLK/ Signals

Because the BCLK/ and CCLK/ signals drive edge-sensitive devices, signal ringing is more likely to cause problems in these signals than in others. Unless you take precautions to reduce the effect of this signal ringing, boards receiving BCLK/ and CCLK/ might misinterpret the ringing as being valid BCLK/ or CCLK/ edges.

Reducing Effects of Signal Ringing in the BCLK/ and CCLK/ Lines

The technique for reducing signal ringing in BCLK/ and CCLK/ is very straightforward. Place boards receiving BCLK/ between the Multibus terminators and the board generating BCLK/. Similarly, place boards receiving CCLK/ between the terminators and the board generating CCLK/. This causes the BCLK/ and CCLK/ lines to be terminated, and reduces the magnitude of the signal ringing in the lines.

Summary of Noise Reduction

Signal-to-signal coupling, ground shifting, or signal ringing can introduce electrical noise into your Multibus signals, but these phenomena do not generally create enough noise to cause your system to fail. Nonetheless, if your system is failing for reasons that you cannot identify, you should modify the configuration of your system to adhere to the following noise reduction guidelines:

- Reduce signal-to-signal coupling and signal ringing in the BCLK/ and CCLK/ lines by following these two steps:

  If CCLK/ is not required in your system, do not drive it on the Multibus line.

  - Place the receivers of a signal between the generator and the Multibus terminators, especially for BCLK/ and CCLK/.

  - Place the whole collection (generators and receivers) as close as possible to the Multibus terminators.

  - If your system requires CCLK/, generate it using a board that drives CCLK/ 180 degrees out of phase with BCLK/.

- Place boards receiving a signal as close as possible to the board generating the signal.

- If your system has many empty slots, try to place the boards near the Multibus terminator, leaving the empty slots near the unterminated end of the backplane.

- When possible, avoid leaving empty slots between boards. This helps reduce the length of current-carrying traces.